

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)

Naoya FUJISAKI)

Serial No.: 09/819,701)

Filed: March 29, 2001)

For: FILE SYSTEM ASSIGNING A)
SPECIFIC ATTRIBUTE TO A FILE,)
A FILE MANAGEMENT METHOD)
ASSIGNING A SPECIFIC ATTRIBUTE)
TO A FILE, AND A STORAGE)
MEDIUM ON WHICH IS RECORDED)
A PROGRAM FOR MANAGING FILES)

Group Art Unit: Unassigned

Examiner: Unassigned

RECEIVED
JUL 25 2001
Group 2100

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55**

*Assistant Commissioner for Patents
Washington, D.C. 20231*

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-358750
Filed: November 27, 2000

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,
STAAS & HALSEY LLP

Date: April 20, 2001

By: _____

James D. Halsey, Jr.
Registration No. 22,729

700 11th Street, N.W., Ste. 500
Washington, D.C. 20001
(202) 434-1500

PATENT OFFICE
JAPANESE GOVERNMENT



This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: November 27, 2000

Application Number: Patent Application
No. 2000-358750

Applicant(s): FUJITSU LIMITED

RECEIVED
JUL 25 2001
Group 2100

RECEIVED
JUL 25 2001
Group 2100

March 9, 2001

Commissioner,
Patent Office Kozo Oikawa

Certificate No. 2001-3017161

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application:

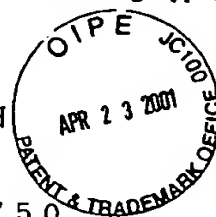
2000年11月27日

出 願 番 号
Application Number:

特願2000-358750

出 願 人
Applicant (s):

富士通株式会社

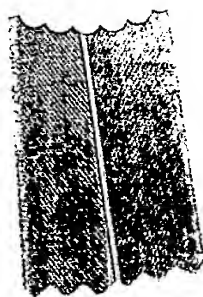


RECEIVED
MAY 25 2001
Group 2100

CERTIFIED COPY OF
PRIORITY DOCUMENT

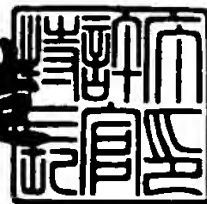


2001年 3月 9日



特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2001-3017161

【書類名】 特許願

【整理番号】 0051465

【提出日】 平成12年11月27日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 12/00

【発明の名称】 ファイルシステムとそのファイルシステムの実現に用い
られるプログラム記録媒体

【請求項の数】 10

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通
株式会社内

 【氏名】 藤崎 直哉

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【代理人】

 【識別番号】 100095072

 【弁理士】

 【氏名又は名称】 岡田 光由

 【電話番号】 03-3807-1818

【選任した代理人】

 【識別番号】 100074848

 【弁理士】

 【氏名又は名称】 森田 寛

【手数料の表示】

 【予納台帳番号】 012944

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9707817

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ファイルシステムとそのファイルシステムの実現に用いられるプログラム記録媒体

【特許請求の範囲】

【請求項 1】 1 つ又は複数のボリュームで構成されるファイルシステムにおいて、

ディレクトリのパス情報とそれに対応付けて指定される 1 つ又は複数の属性データとで構成されるポリシデータを入力する手段と、

上記ポリシデータに従ってディレクトリに割り付けた属性データについては、その配下のディレクトリに継承させる形で割り付けるとともに、継承を要求されない属性データについては、上記ポリシデータの持つ属性データを優先する形でディレクトリに割り付けることで、各ディレクトリに属性データを割り付ける手段とを備えることを、

特徴とするファイルシステム。

【請求項 2】 請求項 1 に記載されるファイルシステムにおいて、

ファイルを配下に持つディレクトリに、その配下のファイルのサイズの合計情報を上記属性データの 1 つとして割り付ける手段を備えることを、

特徴とするファイルシステム。

【請求項 3】 請求項 1 又は 2 に記載されるファイルシステムにおいて、

ファイルデータがアーカイブファイルに格納される場合に、隠しファイルを作成して、そこに上記ポリシデータを格納し、それをアーカイブファイルに格納する手段を備えることを、

特徴とするファイルシステム。

【請求項 4】 請求項 3 に記載されるファイルシステムにおいて、

アーカイブファイルに格納されるファイルデータが再格納される場合に、上記隠しファイルに格納される上記ポリシデータを読み込んで登録する手段を備えることを、

特徴とするファイルシステム。

【請求項 5】 請求項 4 に記載されるファイルシステムにおいて、

上記再格納に応じて発行されるディレクトリの作成要求で指定される親ディレクトリに対応付けて設けられて、上記ポリシデータの持つディレクトリパス情報を検索する必要があるのか否かを示す情報を記録する手段を備えることを、

特徴とするファイルシステム。

【請求項 6】 請求項 4 に記載されるファイルシステムにおいて、

上記再格納に応じて発行されるディレクトリの作成要求で指定される親ディレクトリに対応付けて設けられて、該ディレクトリパス情報をどこまで検索したのかについて示す情報を管理する制御表へのポインタ情報を記録する手段を備えることを、

特徴とするファイルシステム。

【請求項 7】 請求項 6 に記載されるファイルシステムにおいて、

上記記録の手段は、上記ディレクトリパス情報を検索する必要がある場合には、上記ポインタ情報を記録しないことを、

特徴とするファイルシステム。

【請求項 8】 請求項 1 ないし 7 のいずれか 1 項に記載されるファイルシステムにおいて、

ファイル操作に従って上記ポリシデータを違反する状態が発生する場合に、上記ポリシデータの持つ該当する上記属性データにその旨を記録する手段を備えることを、

特徴とするファイルシステム。

【請求項 9】 請求項 1 ないし 8 のいずれか 1 項に記載されるファイルシステムにおいて、

ポリシ違反しているファイル又はディレクトリのポリシを回復させ、かつ最適化を図る手段を備えることを、

特徴とするファイルシステム。

【請求項 10】 1 つ又は複数のボリュームで構成されるファイルシステムの実現に用いられるプログラムを記録したプログラム記録媒体であって、

ディレクトリのパス情報とそれに対応付けて指定される 1 つ又は複数の属性データとで構成されるポリシデータを入力する処理と、

上記ポリシデータに従ってディレクトリに割り付けた属性データについては、その配下のディレクトリに継承させる形で割り付けるとともに、継承を要求されない属性データについては、上記ポリシデータの持つ属性データを優先する形でディレクトリに割り付けることで、各ディレクトリに属性データを割り付ける処理とをコンピュータに実行させるプログラムを記録したことを、

特徴とするプログラム記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、1つ又は複数のボリュームで構成されるファイルシステムと、そのファイルシステムの実現に用いられるプログラムを記録したプログラム記録媒体とに関し、特に、通常のファイル操作インタフェースの性能を損なうことなく、また、通常のファイルシステムとの互換性を実現しつつ、固有の属性を付加することを實現するファイルシステムと、そのファイルシステムの実現に用いられるプログラムを記録したプログラム記録媒体とに関する。

【0002】

近年のコンピュータシステムの普及とともに、あらゆる分野でコンピュータが利用されている。特に会社などの組織のコンピュータシステムにおいては、情報を効率良く共有するために、情報を集中管理するサーバを設けている。

【0003】

また、昨今、インターネットやイントラネットの普及につれて、予め見積もられていた組織自身が収集・作成した情報の他に、電子メールやWEBファイルのダウンロードなどによる個人の情報が予測外に蓄積され、これに応じて、否応なしに、サーバに情報を保存するディスク装置が次々と追加されているのが実情である。

【0004】

現状のように、サーバに蓄えられている様々な情報がディスク装置群に混沌と保存されていると、あるディスク装置の故障により、何の情報が失われたのかを把握するのが困難である。

【0005】

このような大量の情報を管理するときに起こる問題点を解決するために、情報の種類により情報をディスク装置に分散配置するというようなポリシーが導入されてきている。

【0006】

このようなポリシーの導入にあたっては、ポリシー処理オーバーヘッドによるシステム性能の低下は避けなければならない、更に、ポリシー管理された情報（ファイルの属性）と、そうでない一般の情報（ファイルの属性）との互換性を考慮しなければならない。

【0007】

【従来の技術】

本出願人が言うポリシーとは、ファイルに固有の属性を与えて、それに従ってある処理を実行することで、ファイルシステムに付加価値を与えることを実現する技術である。

【0008】

従来技術のファイルシステムで実現されているポリシーとしては、UNIXのquota(クォータ)がある。このUNIXのquotaは、ユーザ毎に利用ディスク容量を制限するというポリシーを実現している。

【0009】

このquotaと組み合わせるアプリケーションは、あるユーザに許可されている利用制限量を超えると、そのユーザに対して、その旨を電子メールで通知したり、管理者からの指示を電子メールで通知することなどを行っている。

【0010】

【発明が解決しようとする課題】

しかしながら、従来技術のファイルシステムで実現されているポリシーとしては、このUNIXのquotaの他にはないというのが実情である。

【0011】

例えば、UNIXのquotaでは、ユーザ毎に利用ディスク容量を制限するというポリシーを実現しているが、ユーザ毎ではなくて、あるグループ毎に利用ディス

ク容量を制限するというポリシーを実現したい場合もある。しかしながら、従来技術に従うファイルシステムでは、このようなポリシーを実現できない。

【 0 0 1 2 】

また、現在のファイルシステムは、情報の増大に伴って、通常、ディスク（ボリューム）を複数備えるマルチボリューム構成を採っている。

【 0 0 1 3 】

このようなマルチボリューム構成を採る場合に、ファイルシステムの管理者（人間）は、アクセス頻度の大きいファイルについては、アクセス速度の速いディスクに格納したというように、あるファイルのあるディスクに格納したいと考えることがある。

【 0 0 1 4 】

しかしながら、従来技術のファイルシステムでは、ファイルをどのディスクに格納するのかについてはファイルシステム自身が決定しており、ファイルシステムの管理者はその決定に関与できず、従って、このような要望に応えるポリシーを実現できない。

【 0 0 1 5 】

また、このようなマルチボリューム構成を採る場合に、ファイルシステムの管理者は、あるファイルのアクセス頻度を測定して、アクセス頻度の大きいファイルについては、アクセス速度の速い別のディスクに移したいと考えることがある。

【 0 0 1 6 】

しかしながら、従来技術のファイルシステムでは、ファイルシステム全体としてのアクセス頻度を測定するという技術については提供されているものの、あるファイルのアクセス頻度やあるファイルグループのアクセス頻度を測定するという技術については提供されておらず、従って、このような要望に応えるポリシーを実現できない。

【 0 0 1 7 】

このように、従来技術のファイルシステムで提供されているポリシーはUNIXのquota だけであり、非常に不十分である。

【 0 0 1 8 】

一方、ポリシの実現にあたっては、ポリシ処理のオーバーヘッドが大きくなりな
いようにしていく必要がある。

【 0 0 1 9 】

UNIXのquota で実現されているポリシでは、ユーザに許可されている利用
制限量を超えると、そのユーザに対して、その旨を電子メールで通知することな
どを行っているが、ファイルシステムの管理者からみると、このポリシの実現に
は、ユーザが電子メールを受信してから不要ファイルを削除するまでの時間を要
することから、ポリシ処理のオーバーヘッドが大きいという問題点がある。

【 0 0 2 0 】

そして、ポリシの実現にあたっては、ファイルシステムで行われるバックアッ
プの処理で、その実現に用いられる属性データが消失されないようにしていく必
要がある。

【 0 0 2 1 】

UNIXのquota で実現されているポリシでは、ポリシ管理されたファイル（
ポリシを実現するための属性データを管理しているファイル）をアーカイブファ
イルにバックアップする場合に、アーカイブファイルはファイルデータのみをバ
ックアップすることから、そのポリシを実現するための属性データが消失してし
まうという問題点がある。

【 0 0 2 2 】

そして、ポリシの実現にあたっては、異なるファイルシステム間での互換性を
持たせる必要がある。

【 0 0 2 3 】

本発明はかかる事情に鑑みてなされたものであって、通常ファイル操作イン
タフェースの性能を損なうことなく、また、通常ファイルシステムとの互換性
を実現しつつ、固有の属性を付加することを実現する新たなファイルシステムの
提供と、そのファイルシステムの実現に用いられるプログラムを記録した新たな
プログラム記録媒体の提供を目的とする。

【 0 0 2 4 】

【課題を解決するための手段】

この目的を達成するために、本発明のファイルシステムでは、1つ又は複数のボリュームで構成されるときにあって、ディレクトリのパス情報とそれに対応付けて指定される1つ又は複数の属性データとで構成されるポリシデータを入力する入力手段と、入力されたポリシデータに従ってディレクトリに割り付けた属性データについては、その配下のディレクトリに継承させる形で割り付けるとともに、継承を要求されない属性データについては、入力されたポリシデータの持つ属性データを優先する形でディレクトリに割り付けることで、各ディレクトリに属性データを割り付ける割付手段とを備えるように構成する。

【0025】

この構成を採るときにあって、ファイルを配下に持つディレクトリに、その配下のファイルのサイズの合計情報を属性データの1つとして割り付ける第2の割付手段を備えることがある。

【0026】

また、ファイルデータがアーカイブファイルに格納される場合に、隠しファイルを作成して、そこに入力されたポリシデータを格納し、それをアーカイブファイルに格納する格納手段と、アーカイブファイルに格納されるファイルデータが再格納される場合に、その隠しファイルに格納されるポリシデータを読み込んで登録する登録手段とを備えることがある。

【0027】

また、ディレクトリの作成要求で指定される親ディレクトリに対応付けて設けられて、ポリシデータの持つディレクトリパス情報をどこまで検索したのかについて示す情報を管理する制御表へのポインタ情報を記録する記録手段を備えることがある。この記録手段は、ポリシデータの持つディレクトリパス情報を検索する必要がある場合には、そのポインタ情報を記録しないことで、ポリシデータの持つディレクトリパス情報を検索する必要があることを表示する。

【0028】

また、ファイル操作に従ってポリシデータを違反する状態が発生する場合に、ポリシデータの持つ該当する属性データにその旨を記録する第2の記録手段を備

えることがある。

【0029】

また、ポリシ違反しているファイル又はディレクトリのポリシを回復させ、かつ最適化を図る回復手段を備えることがある。

【0030】

このように構成される本発明のファイルシステムでは、割付手段は、入力手段がディレクトリのパス情報とそれに対応付けて指定される1つ又は複数の属性データとで構成されるポリシデータを入力すると、入力されたポリシデータに従ってディレクトリに割り付けた属性データについては、その配下のディレクトリに継承させる形で割り付けるとともに、継承を要求されない属性データについては、入力されたポリシデータの持つ属性データを優先する形でディレクトリに割り付けることで、各ディレクトリに属性データを割り付ける。

【0031】

例えば、図1に示すように、3つの属性データを持つポリシデータを入力するときに、第1番目と第3番目の属性データが継承特性を持たず、第2番目の属性データが継承特性を持つ場合にあって、

(／root／A) : (a, b, c)

(／root／A／B／D) : (d, 指定なし, e)

というポリシデータを入力するときには、①ディレクトリAに対して(a, b, c)という属性データを割り付け、②ディレクトリBに対して(a, b, c)という属性データを割り付け、③ディレクトリCに対して(d, b, e)という属性データを割り付けるのである。

【0032】

このような属性データの割り付け機能を持つことから、本発明のファイルシステムによれば、例えば、ポリシデータの持つ属性データの1つとしてボリューム番号を定義することで、ファイルシステムの管理者は、ファイルの格納先のディレクトリの指定に応じて、そのファイルを自分の希望するディスクに格納することができるようになる。

【0033】

また、例えば、ポリシデータの持つ属性データの1つとしてquotaを定義することで、ファイルシステムの管理者は、あるディレクトリ配下のファイルの利用ディスク容量を制限できるようになるので、あるグループ毎に利用ディスク容量を制限することができるようになる。

【 0 0 3 4 】

また、例えば、ポリシデータの持つ属性データの1つとしてこういう統計情報を収集するという命令を定義することで、ファイルシステムの管理者は、あるディレクトリ配下のファイルやファイルグループを収集対象として、そのファイルやそのファイルグループの統計情報を収集することができるようになる。

【 0 0 3 5 】

このようにして、本発明のファイルシステムによれば、ディレクトリパス情報に対応付ける形でファイルに固有の属性データを与えることが可能になり、これにより様々な付加価値を与えることができるようになる。そして、このとき、属性データの機能に応じてその継承方法を変えるようにしているので、個々のポリシ機能の一貫性を維持することができるようになる。

【 0 0 3 6 】

この構成を採るときにあって、第2の割付手段は、ファイルを配下に持つディレクトリに、その配下のファイルのサイズの合計情報を属性データの1つとして割り付ける。

【 0 0 3 7 】

この機能が用意されることで、例えば、renameシステムコールによりディスクに跨がって移動するファイルの総合データサイズをディレクトリのみを検索することで求めることが可能となり、ファイルのメタデータのアクセスのオーバーヘッドをなくすことができるようになる。

【 0 0 3 8 】

一方、この構成を採るときにあって、格納手段は、ファイルデータがアーカイブファイルに格納される場合に、隠しファイルを作成して、そこに入力されたポリシデータを格納し、それをアーカイブファイルに格納する処理を行う。

【 0 0 3 9 】

アーカイブファイルはファイルデータのみをバックアップすることから、この格納手段が用意されないと、バックアップの際に、入力されたポリシデータは失われてしまう。そこで、ポリシデータのバックアップを行う格納手段というインタフェースを設けるのである。

【 0 0 4 0 】

このバックアップインタフェース機能が用意されることで、ファイルシステム固有のポリシデータが取り出されて、隠しファイルの形で、ファイルデータの1つとしてアーカイブファイルにバックアップされることになるので、従来のアーカイブファイルのデータフォーマットの仕様を変えることなく、ファイルシステムに依存したポリシデータを保持することができるようになる。

【 0 0 4 1 】

一方、この構成を採るときにあって、登録手段は、アーカイブファイルに格納されるファイルデータが再格納される場合に、その隠しファイルに格納されるポリシデータを読み込んで登録する処理を行う。

【 0 0 4 2 】

上述の格納手段の処理に従って、ファイルシステム固有のポリシデータをアーカイブファイルに単にバックアップしても、そのポリシデータは隠しファイルとして存在するものの、バックアップ元のファイル自身のポリシデータは失われてしまう。そこで、ポリシデータのリストアを行う登録手段というインタフェースを設けるのである。

【 0 0 4 3 】

このリストアインタフェース機能が用意されることで、アーカイブファイルの隠しファイルに格納されるポリシデータが取り出されて、そのポリシデータがファイルシステムに登録設定されることになるので、従来のアーカイブファイルのデータフォーマットの仕様を変えることなく、ファイルシステムに依存したポリシデータを持つファイルを、別のファイルシステムに復元することができるようになる。

【 0 0 4 4 】

一方、この構成を採るときにあって、ディレクトリの作成要求で指定される親

ディレクトリに対応付けて設けられる記録手段は、ポリシデータの持つディレクトリパス情報のどこまでを検索したのかについて示す情報を管理する制御表へのポインタ情報を記録するとともに、そのディレクトリパス情報を検索する必要がある場合には、そのポインタ情報を記録しないことで、ポリシデータの持つディレクトリパス情報を検索する必要があるのか否かを表示するとともに、検索する必要がある場合には、ディレクトリパス情報のどこから検索したらよいのかについて表示する処理を行う。

【 0 0 4 5 】

アーカイブファイルに格納されるファイルデータがファイルシステムに再格納されるときに、ポリシデータの指定されるディレクトリに対してはポリシデータの持つ属性データを割り付けていく必要がある。

【 0 0 4 6 】

このとき、この記録手段が用意されることで、不要なパス検索処理を避けることができるようになるとともに、既に検索したパス部分の重複する検索を避けることができるようになることで、パス検索処理のオーバーヘッドを大幅に削減することができるようになる。

【 0 0 4 7 】

一方、この構成を採るときにあって、第 2 の記録手段は、ファイル操作に従ってポリシデータに違反する状態が発生する場合に、ポリシデータの持つ該当する属性データにその旨を記録する処理を行う。

【 0 0 4 8 】

例えば、ボリュームを割り当てる際に、1つのファイルシステム上でポリシデータの指定するボリュームに空き領域がない場合に、他の空き領域のあるボリュームに割り当てることでアクセスエラーが発生しないようにすることが好ましい。

【 0 0 4 9 】

このとき、この第 2 の記録手段が用意されることで、ポリシデータに叶わなかったファイルをポリシ違反して記録することができるようになることで、ポリシ実現処理のオーバーヘッドをなくすことができるようになるとともに、書込エラー

の発生を押さえて、ポリシーの如何に関わらず汎用のUNIXファイルシステムの仕様を遵守することができるようになる。

【 0 0 5 0 】

UNIXのquota で実現されているポリシーでは、ユーザに許可されている利用制限量を超えると、そのユーザに対して、その旨を電子メールで通知することなどを行うために、ポリシー処理のオーバーヘッドが大きいという問題点がある。

【 0 0 5 1 】

これに対して、本発明によれば、第2の記録手段が用意されることで、ポリシー違反の発生を記録する機構が用意されているので、利用制限量を超えた形での一時的な利用が可能になることで、ポリシー処理のオーバーヘッドを削減することができるようになる。

【 0 0 5 2 】

一方、この構成を採るときにあって、回復手段は、ポリシー違反しているファイル／ディレクトリに対して、その最適化を図りつつポリシーを回復させる処理を行う。

【 0 0 5 3 】

例えば、ボリュームの空き領域がなくて、ポリシーと異なるボリュームが割り当てられた後、当初の目的のボリュームに空き領域ができたり、当初のポリシーに新たにボリュームを追加することなどにより、ポリシー回復が見込まれる場合には、現ボリュームから、そのようなポリシーに沿うボリュームにデータを移動させ、かつ、そのデータの配置が最適であるようにすることがデータマネジメント上望ましい。

【 0 0 5 4 】

このとき、この回復手段が用意されることで、記録しておいたポリシー違反リストから、ポリシー回復対象のファイル／ディレクトリを瞬時に探し出し、ボリュームの空き領域などのシステムの状態から判断して、ポリシーの回復が可能ならば、その回復と最適化を図る処理を行うことができるようになる。

【 0 0 5 5 】

【発明の実施の形態】

以下、実施の形態に従って本発明を詳細に説明する。

【 0 0 5 6 】

図 2 に、本発明のファイルシステム（1つのファイルシステムで構成されることもあるし、複数のファイルシステムで構成されることもある）により実現されるポリシーの指定処理の一例を図示する。

【 0 0 5 7 】

この図に示すように、本発明のファイルシステムでは、例えば、ディレクトリ X にポリシー(X,1) が指定されると、そのポリシー(X,1) を図中に示すpolicy(X,1) が区画するディレクトリに継承させ、ディレクトリ X にポリシー(X,2) が指定されると、そのポリシー(X,2) を図中に示すpolicy(X,2) が区画するディレクトリに継承させ、ディレクトリ A にポリシー(A,1) が指定されると、そのポリシー(A,1) を図中に示すpolicy(A,1) が区画するディレクトリに継承させ、ディレクトリ A にポリシー(A,2) が指定されると、そのポリシー(A,2) を図中に示すpolicy(A,2) が区画するディレクトリに継承させるというように、ディレクトリに対してポリシーが指定される場合に、原則として、そのポリシーをその配下のディレクトリに継承させるとともに、継承が要求されないポリシーについては、その配下のディレクトリに対して指定されるポリシーを優先する形で、ディレクトリに対してポリシーを指定するように処理する。

【 0 0 5 8 】

図 3 に、このようなポリシー指定処理を実行する本発明のファイルシステムの一実施形態例を図示する。

【 0 0 5 9 】

図中、1 は本発明のファイルシステム、2 は本発明のファイルシステム 1 が備えるポリシー制御モジュール、3 は本発明のファイルシステム 1 が備えるパス検索モジュール、10 はユーザ空間である。

【 0 0 6 0 】

ここで、ポリシー制御モジュール 2 やパス検索モジュール 3 を実現するプログラムは、計算機が読み取り可能な半導体メモリなどの適当な記録媒体に格納することができる。

【 0 0 6 1 】

ポリシーの指定処理にあたって、本発明のファイルシステム 1 の管理者は、図中の A で示すように、処理対象となるポリシーデータを登録する。このとき登録されるポリシーデータは、図 1 に示したように、ディレクトリのパス情報（目的ディレクトリ）と、それに対応付けて指定される 1 つ又は複数の属性データ（機能データ）とで構成されている。

【 0 0 6 2 】

このポリシーデータの登録を受けて、本発明のファイルシステム 1 は、図中の B で示すように、登録されたポリシーデータを入力し、それが持つ目的ディレクトリを検索して、目的ディレクトリが見つからない場合には、パス検索モジュール 3 を起動し、目的ディレクトリが見つかった場合には、ポリシー制御モジュール 2 を起動する。

【 0 0 6 3 】

このようにして起動されると、ポリシー制御モジュール 2 は、ポリシーデータの持つ属性データ（機能データ）毎に、以下の処理を繰り返す。

【 0 0 6 4 】

すなわち、先ず最初に、メタデータから親ディレクトリの持つ処理対象の属性データを取得して、それと登録されたポリシーデータの持つ処理対象の属性データとを比較する。そして、属性データ毎に定義される継承属性に従って、親ディレクトリの属性データを継承するのか否かを判定して、親ディレクトリの属性データを継承する場合には、目的ディレクトリにその親ディレクトリの属性データを割り付け、一方、親ディレクトリの属性データを継承する必要がなければ、目的ディレクトリに指定された属性データを目的ディレクトリに割り付ける。

【 0 0 6 5 】

このようにして、本発明のファイルシステム 1 では、ディレクトリに対してポリシーが指定される場合に、原則として、そのポリシーをその配下のディレクトリに継承させるとともに、継承が要求されないポリシーについては、その配下のディレクトリに対して指定されるポリシーを優先する形で、ディレクトリに対してポリシーを指定するように処理するのである。

【 0 0 6 6 】

このような属性データの割り付け機能を持つことから、本発明のファイルシステム 1 によれば、例えば、ポリシーデータの持つ属性データの 1 つとしてボリューム番号を定義することで、本発明のファイルシステム 1 の管理者は、ファイルの格納先のディレクトリの指定に応じて、そのファイルを自分の希望するディスクに格納することができるようになる。

【 0 0 6 7 】

また、例えば、ポリシーデータの持つ属性データの 1 つとして quota を定義することで、本発明のファイルシステム 1 の管理者は、あるディレクトリ配下のファイルの利用ディスク容量を制限できるようになるので、あるグループ毎に利用ディスク容量を制限することができるようになる。

【 0 0 6 8 】

また、例えば、ポリシーデータの持つ属性データの 1 つとしてこういう統計情報を収集するという命令を定義することで、本発明のファイルシステム 1 の管理者は、あるディレクトリ配下のファイルやファイルグループを収集対象として、そのファイルやそのファイルグループの統計情報を収集することができるようになる。

【 0 0 6 9 】

本発明のファイルシステム 1 もまた、本発明を具備しないファイルシステムと同様に、ファイルデータをアーカイブファイルにバックアップしなければならないことが起こる。

【 0 0 7 0 】

アーカイブファイルはファイルデータのみをバックアップすることから、従来技術に従っていると、バックアップの際に、ディレクトリに割り付けたポリシーデータが失われてしまうことになる。

【 0 0 7 1 】

そこで、本発明のファイルシステム 1 では、図 4 に示すように、ファイルデータをアーカイブファイル 2 0 にバックアップする際に、隠しファイル（図中の st ab-file）を作成して、そこにポリシーデータを格納し、それをアーカイブファイル

20にバックアップするとともに、アーカイブファイル20にバックアップされたファイルデータをリストアする際に、その隠しファイルにバックアップされたポリシデータをリストアするという処理を行うポリシ・バックアップ・リストア・インタフェースモジュール4を備える構成を採っている。

【0072】

次に、図5に従って、このポリシ・バックアップ・リストア・インタフェースモジュール4の実行する処理について詳細に説明する。

【0073】

本発明のファイルシステム1の管理者は、ファイルデータをアーカイブファイル20にバックアップする際に、ポリシデータのバックアップ命令を発行する（図中のA1）。

【0074】

このバックアップ命令を受け取ると、ポリシ・バックアップ・リストア・インタフェースモジュール4は、それをポリシ制御モジュール2に渡し（図中のA2）、これを受けて、ポリシ制御モジュール2は、管理するポリシデータを取り出して、それをポリシ・バックアップ・リストア・インタフェースモジュール4に渡す（図中のA3）。

【0075】

このポリシデータを受けて、ポリシ・バックアップ・リストア・インタフェースモジュール4は、その渡されるポリシデータを発行されたバックアップ命令に渡し（図中のA4）、これを受けて、発行されたバックアップ命令は、隠しファイル（図中のstab-file）を作成して、そこに受け取ったポリシデータを格納する（図中のA5）

最後に、発行されたバックアップ命令は、ファイルシステムのバックアップを隠しファイルとともにアーカイブファイル20に保存して（図中のA6）、処理を終了する。

【0076】

一方、本発明のファイルシステム1の管理者は、アーカイブファイル20にバックアップされたファイルデータをリストアする際に、ポリシデータのリストア

命令を発行する（図中のB1）。

【0077】

この発行されたリストア命令は、アーカイブファイル20に格納される隠しファイル（図中のstab-file）から、ポリシデータを抽出して、それをポリシ・バックアップ・リストア・インタフェースモジュール4に渡す（図中のB2）。

【0078】

このポリシデータを受けて、ポリシ・バックアップ・リストア・インタフェースモジュール4は、発行されたリストア命令と受け取ったポリシデータとをポリシ制御モジュール2に渡し（図中のB3）、これを受けて、ポリシ制御モジュール2は、発行されたリストア命令に従って、受け取ったポリシデータを登録／管理して、その旨をポリシ・バックアップ・リストア・インタフェースモジュール4に返す（図中のB4）。

【0079】

これを受けて、ポリシ・バックアップ・リストア・インタフェースモジュール4は、発行されたリストア命令に対してポリシデータのリストアが終了した旨を渡し（図中のB5）、これを受けて、発行されたリストア命令は、隠しファイルを除いて、アーカイブファイル20のデータを本発明のファイルシステム1にリストアして（図中のB6）、処理を終了する。

【0080】

このようにして、本発明のファイルシステム1では、ポリシ・バックアップ・リストア・インタフェースモジュール4が用意されることで、ファイルシステム固有のポリシデータが取り出されて、隠しファイルの形で、ファイルデータの1つとしてアーカイブファイル20にバックアップされることになるので、従来のアーカイブファイルのデータフォーマットの仕様を変えることなく、ファイルシステムに依存したポリシデータを保持することができるようになる。

【0081】

そして、アーカイブファイル20の隠しファイルに格納されるポリシデータが取り出されて、そのポリシデータがファイルシステムに登録設定されることになるので、従来のアーカイブファイルのデータフォーマットの仕様を変えることな

く、ファイルシステムに依存したポリシデータを持つファイルを、別のファイルシステムに復元することができるようになる。

【 0 0 8 2 】

アーカイブファイル 2 0 にバックアップされたファイルデータをリストアする場合には、メイク・ディレクトリ・リクエスト命令 (mkdir-request) が発行されるので、本発明のファイルシステム 1 は、本発明を具備しないファイルシステムと同様に、このメイク・ディレクトリ・リクエスト命令に応答してディレクトリを作成していった、それに続けてファイル・クリエイト命令が発行される場合には、その作成したディレクトリにファイルデータをリストアしていくことになる。

【 0 0 8 3 】

この処理を行うことに加えて、本発明のファイルシステム 1 では、ポリシデータの持つ属性データを対となるディレクトリにリストアする必要があるので、メイク・ディレクトリ・リクエスト命令が発行されるときに、そのメイク・ディレクトリ・リクエスト命令で指定されるディレクトリパスがポリシデータの持つディレクトリパス情報（隠しファイルからリストアされるものを使う）と一致するの否かを検索して、一致する場合には、それに続けて送られてくるポリシデータの持つ属性データを、そのメイク・ディレクトリ・リクエスト命令により作成されたディレクトリにリストアしていく必要がある。

【 0 0 8 4 】

メイク・ディレクトリ・リクエスト命令が発行されると、本発明を具備しないファイルシステムと同様に、本発明のファイルシステム 1 に対して、目的ディレクトリ名（作成対象のディレクトリ名）と、カレントディレクトリ（その目的ディレクトリの親ディレクトリとなるもの）の i ノード番号とが渡される。

【 0 0 8 5 】

これから、この検索処理では、図 6 に示すように、カレントディレクトリと目的ディレクトリ名との対データが、ポリシデータの持つディレクトリパス情報に一致するの否かを検索する必要があり、大変時間のかかる作業となる。

【 0 0 8 6 】

そこで、本発明のファイルシステム 1 では、図 7 に示すように、ポリシデータの持つディレクトリパス情報のどこまでの検索を終えているのかを指し示すチェックポイント（逆に言うと、パス検索処理の開始ポイントを指し示している）を管理するパス検索表 5 を設けて、メイク・ディレクトリ・リクエスト命令で指定されるカレントディレクトリの i ノード番号の指すメタデータに、そのパス検索表 5 のどのチェックポイントを参照すればよいのかを指すチェックインデックスを登録するとともに、検索を行う必要がないときには、このチェックインデックスを登録しないようにするという構成を採っている。

【 0 0 8 7 】

図 7 の例で説明するならば、「home」の配下には「dom1」と「dom2」という 2 つのディレクトリが登録されているが、この 2 つのディレクトリの検索を既に終えている場合（すなわち、この 2 つのディレクトリの作成を終えている場合）には、この後に「home」をカレントディレクトリとするメイク・ディレクトリ・リクエスト命令が発行されても、ポリシデータの持つディレクトリパス情報を検索する必要がないので、図 8 に示すように、「home」の属性を管理するメタデータには、チェックインデックスが登録されていないことで検索処理を行う必要のないことが記録されている。

【 0 0 8 8 】

このことから分かるように、チェックインデックスの登録の有無を判断することで検索をする必要があるのか否かを直ちに判断できるようになるので、不要なパス検索処理を避けることができるようになる。

【 0 0 8 9 】

そして、「dom1」の配下には「grp1」と「grp2」と「grp3」という 3 つのディレクトリが登録されているが、この 3 つの全てのディレクトリの検索を未だ終えていない場合（すなわち、3 つの全てのディレクトリの作成を終えていない場合）には、図 8 に示すように、「dom1」の属性を管理するメタデータには、「dom1」までの検索を終えていることを指し示すチェックポイントの参照を指示するチェックインデックスが登録されていることで検索処理を行う必要のあることが記録されている。

【 0 0 9 0 】

このときには、そのチェックポイントの指す次のディレクトリと照合を行うことで、メイク・ディレクトリ・リクエスト命令で指定される目的ディレクトリ名がポリシデータの持つディレクトリパス情報に登録されているのか否かをチェックする処理を行うことになる。

【 0 0 9 1 】

このことから分かるように、チェックインデックスの指すチェックポイントに従って、既に検索したパス部分（既に作成したパス部分）の重複する検索を避けることができるようになることで、パス検索処理のオーバーヘッドを大幅に削減することができるようになる。

【 0 0 9 2 】

なお、例えば「dom1」の属性を管理するメタデータに登録されているチェックインデックスはあくまで1つであり、そのチェックインデックスの指すチェックポイントが例えば「dom1/grp1」を指しているときには、「grp2」や「grp3」が目的ディレクトリ名として指定されることがあるので、「grp1」が目的ディレクトリ名として指定されない場合には、そのチェックポイントの指し示すエントリーを上下に探索しながら、目的ディレクトリ名がポリシデータの持つディレクトリパス情報に登録されているのか否かをチェックすることになる。

【 0 0 9 3 】

また、例えば「home」の検索を終えた時点で、その配下となる「dom1」の属性を管理するメタデータと、その配下となる「dom2」の属性を管理するメタデータには、「home」の属性を管理するメタデータに登録されていたチェックインデックスが登録されることで、どこまでの検索を終えているのかを指し示すチェックポイントの引き継ぎが行われることになる。

【 0 0 9 4 】

次に、図9の処理フローに従って、メイク・ディレクトリ・リクエスト命令が発行されるときに実行する処理について詳細に説明する。

【 0 0 9 5 】

メイク・ディレクトリ・リクエスト命令が発行されると、本発明のファイルシステム1は、図9の処理フローに示すように、先ず最初に、ステップ1で、そのメイク・ディレクトリ・リクエスト命令を受け取る。

【0096】

続いて、ステップ2で、メイク・ディレクトリ・リクエスト命令で指定される目的ディレクトリの名前と、その親ディレクトリのノード番号とを取得し、更に、そのノード番号から親ディレクトリの属性データ（iノード番号）をメタデータから取得する。そして、その取得した属性データからチェックインデックスが登録されているのか否かを判断するとともに、チェックインデックスが登録されているときには、そのチェックインデックスの指すチェックポイントを取得する。

【0097】

続いて、ステップ3で、チェックインデックスの登録の有無に従ってパス検索の有無を判断して、チェックインデックスが登録されていないことでパス検索を行う必要がないことを判断するときには、発行されたメイク・ディレクトリ・リクエスト命令で指定されるパスについてはポリシデータが指定されていないことを判断して、次のメイク・ディレクトリ・リクエスト命令を待つべくステップ1に戻る。

【0098】

一方、ステップ3で、チェックインデックスが登録されていることでパス検索を行う必要があることを判断するときには、ステップ4に進んで、チェックインデックスの指すパス検索表5のチェックポイントを参照し、続くステップ5で、その参照したチェックポイントに従って対象パス検索データがあるのか否かを判断して、対象パス検索データがないことを判断するときには、ステップ6に進んで、同じノード番号のパス検索データを検索してから、次に説明するステップ7に進む。

【0099】

一方、ステップ5で、対象パス検索データがあることを判断するときには、ステップ7に進んで、そのデータの中のパスチェック開始ポイントからポリシ指定

パス名を抽出して、それとメイク・ディレクトリ・リクエスト命令で指定される目的ディレクトリ名とを比較する。

【0100】

続いて、ステップ8で、この比較により、目的ディレクトリ名がポリシ指定パス名であると同定できない場合には、発行されたメイク・ディレクトリ・リクエスト命令で指定されるパスについてはポリシデータが指定されていないことを判断して、次のメイク・ディレクトリ・リクエスト命令を待つべくステップ1に戻る。

【0101】

一方、目的ディレクトリ名がポリシ指定パス名であると同定できる場合には、ステップ9に進んで、目的ディレクトリの属性データにポリシを設定し、続くステップ10で、対象パス検索データのパスチェック開始ポイントを更新（チェックポイントを進める）してから、ステップ1に戻る。

【0102】

このようにして、本発明のファイルシステム1によれば、メイク・ディレクトリ・リクエスト命令で指定されるディレクトリパスがポリシデータの持つディレクトリパス情報と一致するのか否かを検索するときに、それを高速に実行できるようになるのである。

【0103】

上述したように、本発明のファイルシステム1によれば、ポリシデータの持つ属性データの1つとしてquotaを定義することで、本発明のファイルシステム1の管理者は、あるディレクトリ配下のファイルの利用ディスク容量を制限できるようになる。

【0104】

この利用ディスク量の制限処理を実現するにあたって、各ディスクのサイズを把握する必要がある。そこで、本発明のファイルシステム1では、図10に示すように、ファイルを配下に持つディレクトリに、その配下のファイルのサイズの合計値を属性データの1つとして割り付けるようにしている。

【0105】

この機能が用意されることで、例えば、renameシステムコールによりディスクに跨がって移動するファイルの総合データサイズをディレクトリのみを検索することで求めることが可能となり、ファイルのメタデータのアクセスのオーバーヘッドをなくすることができるようになる。

【0106】

この機能は、具体的には、図11に示すように、①OPENコマンドが発行されると、目的ファイルの名前と、その親ディレクトリのiノード番号を用いて、親ディレクトリの属性データ（ファイルサイズの合計値など）をメタデータから探し出し、②ACCESS/CLOSEコマンドが発行されると、目的ファイルの属性データ（ファイルサイズや変更日時など）を更新し、③それと同時に、親ディレクトリの属性データの1つとして管理されるファイルサイズの合計値を更新することで実現されることになる。

【0107】

ファイルシステムに対しては様々なファイル操作が実行されることになるが、このファイル操作に従って、割り付けられたポリシデータに違反する状態が発生することが起こる。

【0108】

本発明のファイルシステム1は、このようなポリシ違反が発生した場合には、ポリシデータの持つ該当する属性データにその旨を記録する処理を行う。

【0109】

例えば、図12に示すようなマルチボリュームのファイルシステムにおいて、ポリシ属性として、ボリューム割当（volume-ID）と、クォータ（quota）と、パス検索要フラグ（チェックインデックス）とが割り付けられている場合にあって、Yディレクトリを「/A/Y」から「/B/Y」へと名前を変更することを要求するrenameシステムコールが発行されるときには、Bディレクトリのポリシ適用の可否の判断と、この判断により要求されるYディレクトリ配下のファイルデータのボリューム割当の「volume-ID = 1」から「volume-ID = 2」への移動と、クォータの計算とを実行しなければならない。

【0110】

この場合、本発明のファイルシステム 1 は、図 1 3 の処理フローに示すように、Y ディレクトリ配下のファイルデータの移動は行わずに、「volume-ID = - 2」という負の値を記録することで、ポリシ違反であることを表示するように処理する。

【 0 1 1 1 】

これにより、本発明のファイルシステム 1 によれば、このrename処理については次のオーバーヘッドで済むことになる。

【 0 1 1 2 】

すなわち、①B ディレクトリのポリシ属性について、パス検索表 5 のインデックス番号から、rename後のパス名Y ディレクトリがポリシを適用すべきか否かを判定することになるが、これは瞬時に可能である。②また、Y ディレクトリ配下のファイルデータの移動は行わずに、「volume-ID = - 2」という負の値を記録することになるが、これも簡単に処理できる。③また、クォータの計算については、Y ディレクトリ配下のファイルのメタデータの検索を行わずに、ディレクトリ検索のみで計算が可能になり、これも簡単に処理できる。

【 0 1 1 3 】

従って、本発明のファイルシステム 1 によれば、rename処理についてのオーバーヘッドを大幅に削減できるようになる。

【 0 1 1 4 】

このrename処理について更に詳細に説明するならば、renameコマンドが発行されると、本発明のファイルシステム 1 は、目的ファイルの名前と、その親ディレクトリのi ノード番号を用いて、親ディレクトリの属性データをメタデータから探し出す。そして、その探し出した目的ファイルの属性にポリシデータがあれば、または、移動先ディレクトリBの属性にポリシデータ（パス検索の有無フラグも含む）があれば、ポリシ制御モジュール 2 に処理を移行させる。

【 0 1 1 5 】

このようにして起動されると、ポリシ制御モジュール 2 は、目的ファイルと移動先ディレクトリの属性データをチェックする。図 1 2 に示す例では、ディレクトリBにパス検索有フラグ（- 4 : 負値）が立っているので、パス検索処理を行

う。図 1 2 に示す例では、この検索有フラグの指すパス検索表 5 のインデックスには対象のパス検索データがないので、パス検索表 5 を検索して対象のパス検索データを得る。

【0 1 1 6】

続いて、パス検索表 5 のパスのチェックポイントから登録されたポリシ指定のパス名を比較する。この比較により、図 1 2 に示す例では、登録されたポリシ指定のパス名の「/B/Y」は、「volume-ID = 2」のボリュームを割り当てることになっていることが分かる。一方、図 1 2 に示す例では、ディレクトリ Y 配下のデータは「volume-ID = 1」にあるのでポリシ違反となり、図中に示すポリシ違反リストに、ディレクトリ Y の i ノード番号や違反機能 IDなどを登録する。

【0 1 1 7】

このようなポリシ違反を記録する構成を採ることで、ポリシ実現処理のオーバーヘッドをなくすことができるようになるとともに、書込エラーの発生を押さえて、ポリシの如何に関わらず汎用の UNIX ファイルシステムの仕様を遵守することができるようになる。

【0 1 1 8】

最後に、図 1 4 に従って、本発明のファイルシステム 1 の備えるポリシ違反の回復機能について説明する。

【0 1 1 9】

この図 1 4 (a) では、ファイル A については、ポリシ通りにボリューム 1 に割り当てられ、ファイル B についても、ポリシ通りにボリューム 2 に割り当てられているものの、ファイル C については、ポリシではボリューム 1 に割り当てられる筈であるが、ボリューム 1 には収まりきらないためにボリューム 2 に跨がって割り当てられていることを想定している。すなわち、ファイル C はポリシ違反の状態にある。

【0 1 2 0】

この状態にあるときに、図 1 4 (b) に示すように、ファイル A が消去（移動）された場合を想定する。本発明のファイルシステム 1 で用意するポリシ回復機能は、記録しておいたポリシ違反リストから、ポリシ回復対象のファイル／ディ

レクトリを探し出すことで、ポリシ回復対象であるファイルCを瞬時に探し出して、ボリュームの空き領域などのシステムの状態から、その探し出したファイルCのポリシが回復可能かどうかを判断する処理を行う。

【0121】

この図14（b）に示す例では、ファイルAが消去（移動）されることでファイルCのポリシが回復可能であることを判断する。

【0122】

本発明のファイルシステム1で用意するポリシ回復機能は、ポリシの回復可能を判断すると、探し出したポリシ回復対象のファイル／ディレクトリを最適化しつつ、そのポリシを回復させる処理を行う。

【0123】

この図14の例の場合で説明するならば、図14（c）に示すように、ボリューム2に格納されているファイルCのデータが最適な形（この場合には連続する形）でボリューム1に配置されるようにと、ボリューム1に格納されているファイルCのデータを移動させてから、図14（d）に示すように、ボリューム2に格納されているファイルCのデータをボリューム1に移動させることで、ファイルCのポリシ違反を回復させる。このとき、記録しておいたポリシ違反リストからファイルCのポリシ違反の記録を削除する処理を行う。

【0124】

このようにして、本発明のファイルシステム1では、このようなポリシ違反の回復処理の機能を用意することで、ポリシを遵守することが可能となるばかりでなく、データのアクセス性能に対しても最適化を図ることが可能になる。

【0125】

【発明の効果】

以上説明したように、本発明のファイルシステムによれば、ディレクトリパス情報に対応付ける形でファイルに固有の属性データを与えることが可能になり、これにより様々な付加価値を与えることができるようになる。そして、このとき、属性データの機能に応じてその継承方法を変えるようにしているので、個々のポリシ機能の一貫性を維持することができるようになる。

【0126】

例えば、ポリシデータの持つ属性データの1つとしてボリューム番号を定義することで、ファイルシステムの管理者は、ファイルの格納先のディレクトリの指定に応じて、そのファイルを自分の希望するディスクに格納することができるようになる。

【0127】

また、例えば、ポリシデータの持つ属性データの1つとしてquota を定義することで、ファイルシステムの管理者は、あるディレクトリ配下のファイルの利用ディスク容量を制限できるようになるので、あるグループ毎に利用ディスク容量を制限することができるようになる。

【0128】

また、例えば、ポリシデータの持つ属性データの1つとしてこういう統計情報を収集するという命令を定義することで、ファイルシステムの管理者は、あるディレクトリ配下のファイルやファイルグループを収集対象として、そのファイルやそのファイルグループの統計情報を収集することができるようになる。

【0129】

そして、本発明のファイルシステムでは、ファイルを配下に持つディレクトリに、その配下のファイルのサイズの合計情報を属性データの1つとして割り付けるという機能を用意することから、例えば、renameシステムコールによりディスクに跨がって移動するファイルの総合データサイズをディレクトリのみを検索することで求めることが可能となり、ファイルのメタデータのアクセスのオーバーヘッドをなくすことができるようになる。

【0130】

そして、本発明のファイルシステムでは、ポリシデータのバックアップとリトスアを行うインタフェースを用意することから、従来のアーカイブファイルのデータフォーマットの仕様を変えることなく、ファイルシステムに依存したポリシデータを保持することができるようになるとともに、従来のアーカイブファイルのデータフォーマットの仕様を変えることなく、ファイルシステムに依存したポリシデータを持つファイルを、別のファイルシステムに復元することができるよ

うになる。

【0131】

そして、本発明のファイルシステムでは、アーカイブファイルに格納されるファイルデータのリストアに合わせてポリシデータをリストアするときに、ポリシデータの指定されるディレクトリであるのか否かを高速に検索できるようにする機能を用意することから、ポリシデータを高速にリストアできるようになる。

【0132】

そして、本発明のファイルシステムでは、ファイル操作に従ってポリシデータを違反する状態が発生する場合に、ポリシデータの持つ該当する属性データにその旨を記録する機能を用意するので、書込エラーの発生を押さえて、ポリシの如何に関わらず汎用のUNIXファイルシステムの仕様を遵守することができるようになる。

【0133】

このようにして、本発明のファイルシステムによれば、ポリシを実現する処理のオーバーヘッドを低減し、マルチボリュームのファイルシステムで発生するrenameなどの各インタフェースのオーバーヘッドとエラーの発生を低減し、さらにアーカイブファイルのデータフォーマットの仕様を変えることなく、一般に普及しているUNIXファイルシステムなどとの互換性を実現しつつポリシ機能を実現することが可能となるので、システム全体の高性能と高信頼のデータマネージメントに寄与するところが多い。

【図面の簡単な説明】

【図1】

本発明の説明図である。

【図2】

本発明のポリシ指定処理の説明図である。

【図3】

本発明のファイルシステムの一実施形態例である。

【図4】

本発明のポリシデータのバックアップ・リストア処理の説明図である。

【図 5】

本発明のポリシデータのバックアップ・リストア処理の説明図である。

【図 6】

本発明で要求される検索処理の説明図である。

【図 7】

本発明で行う検索処理の説明図である。

【図 8】

パス検索表の説明図である。

【図 9】

本発明で行う検索処理の説明図である。

【図 1 0】

本発明で管理する属性データの説明図である。

【図 1 1】

本発明で行うファイルサイズ合計値の管理処理の説明図である。

【図 1 2】

本発明で行うポリシ違反処理の説明図である。

【図 1 3】

本発明で行うポリシ違反処理の説明図である。

【図 1 4】

本発明で行うポリシ違反回復処理の説明図である。

【符号の説明】

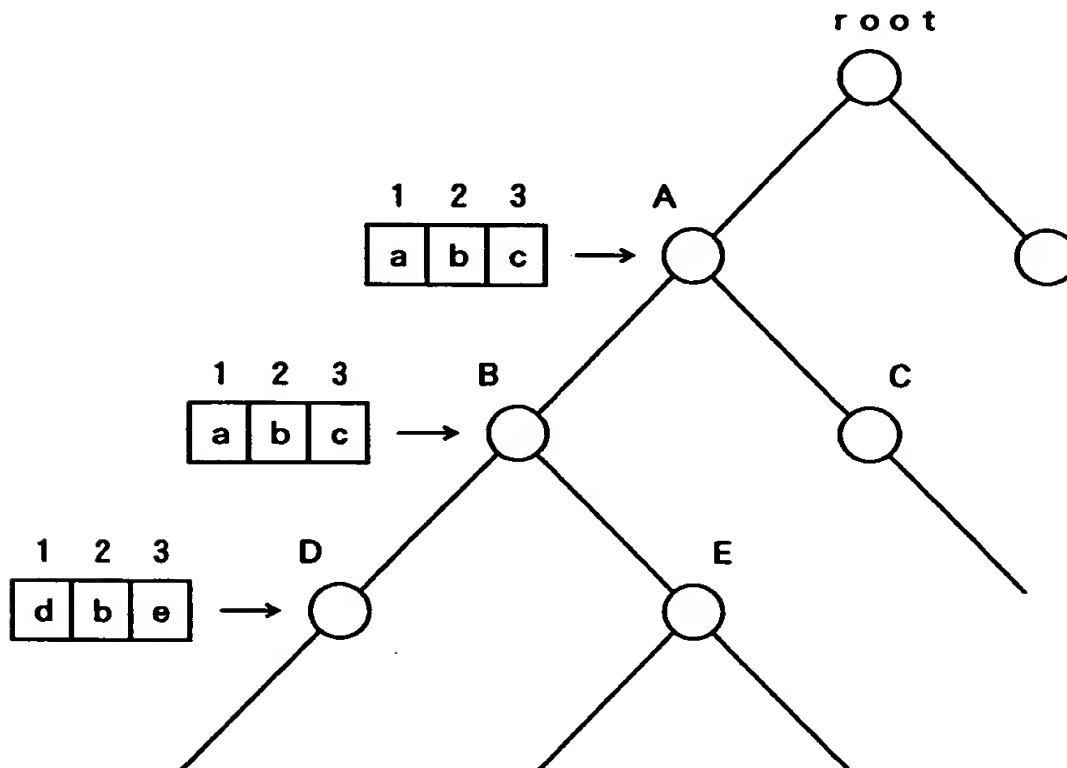
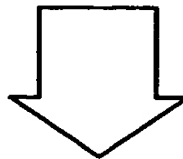
- 1 ファイルシステム
- 2 ポリシ制御モジュール
- 3 パス検索モジュール
- 4 ポリシ・バックアップ・リストア・インタフェースモジュール
- 5 パス検索表
- 1 0 ユーザ空間
- 2 0 アーカイブファイル

【書類名】 図面

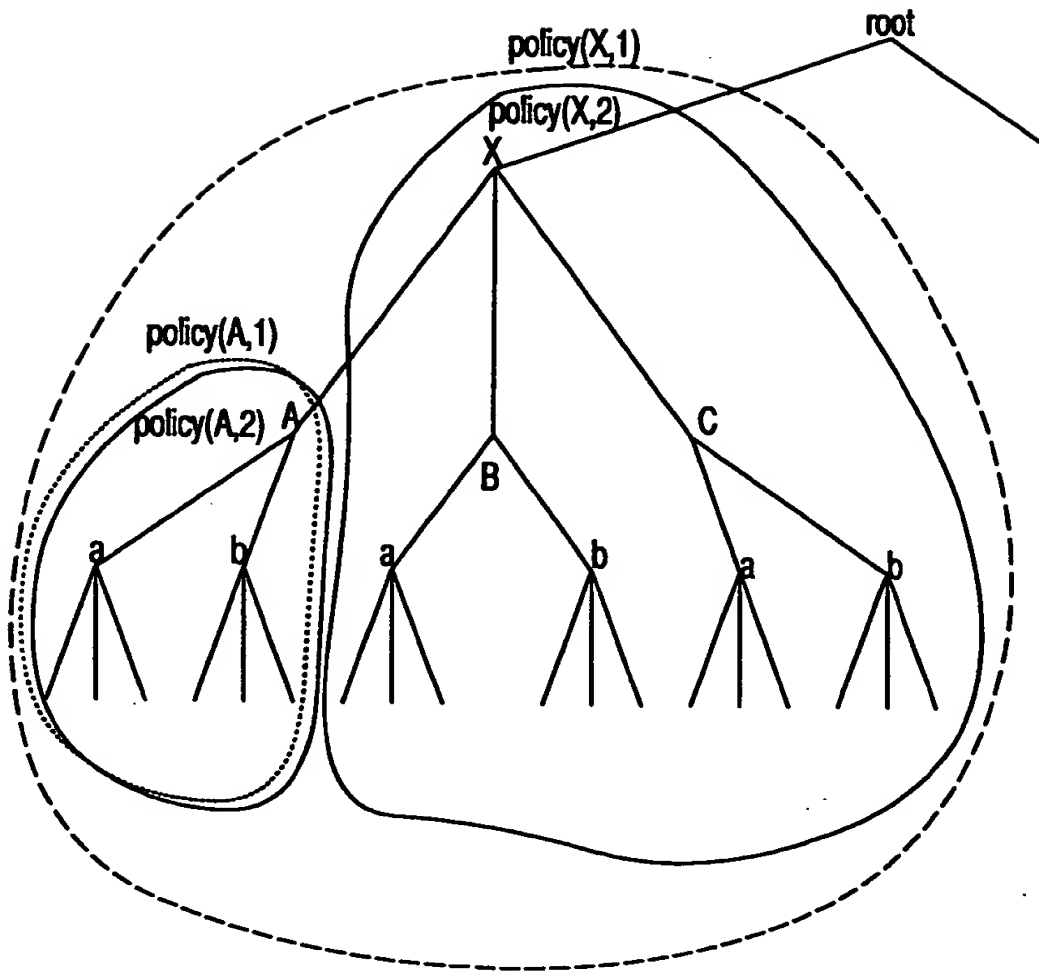
【図1】

入力ポリシデータ

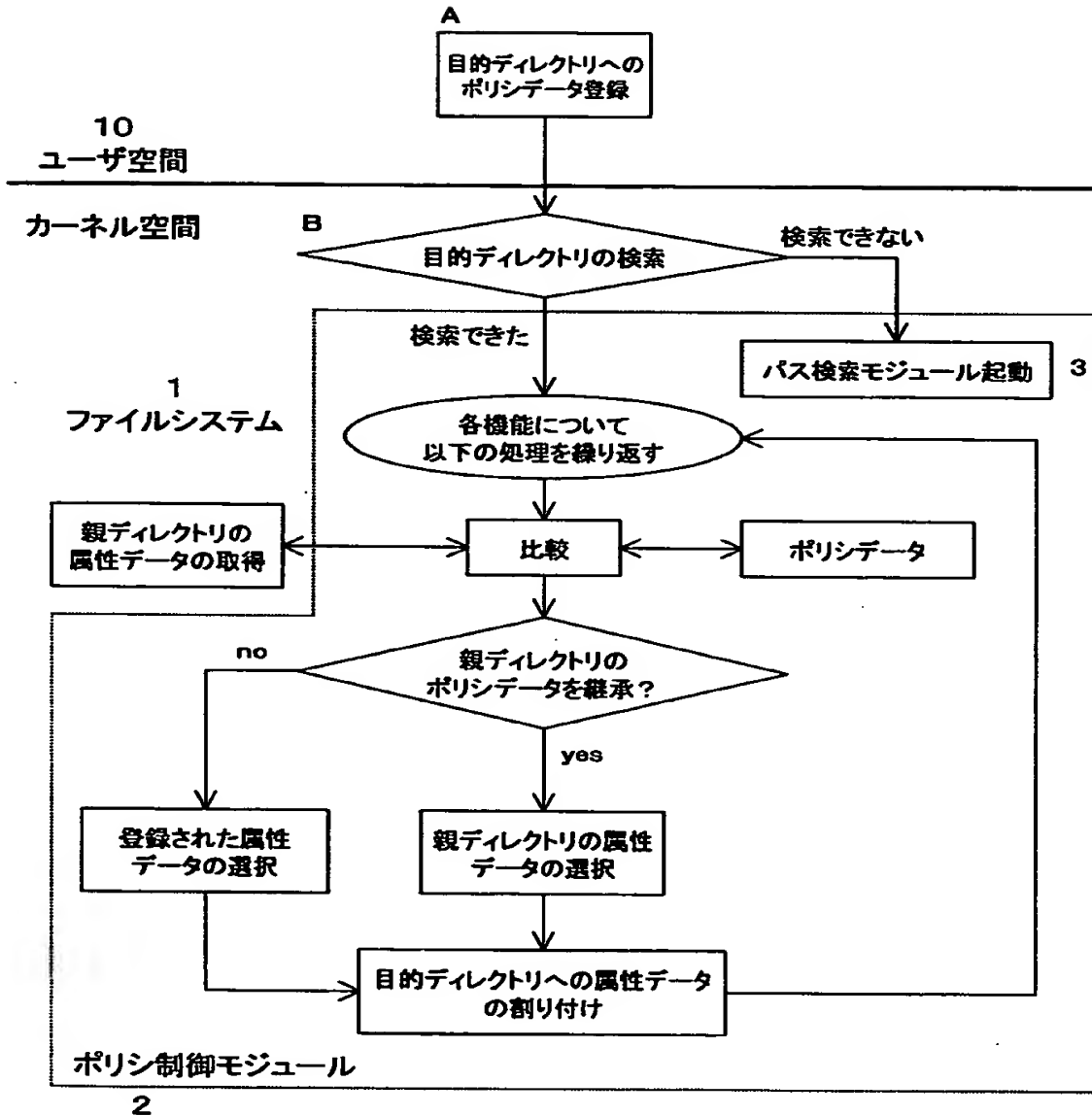
ディレクトリパス情報	属性データ		
	1	2	3
/root/A	a	b	c
/root/A/B/D	d		e
⋮			



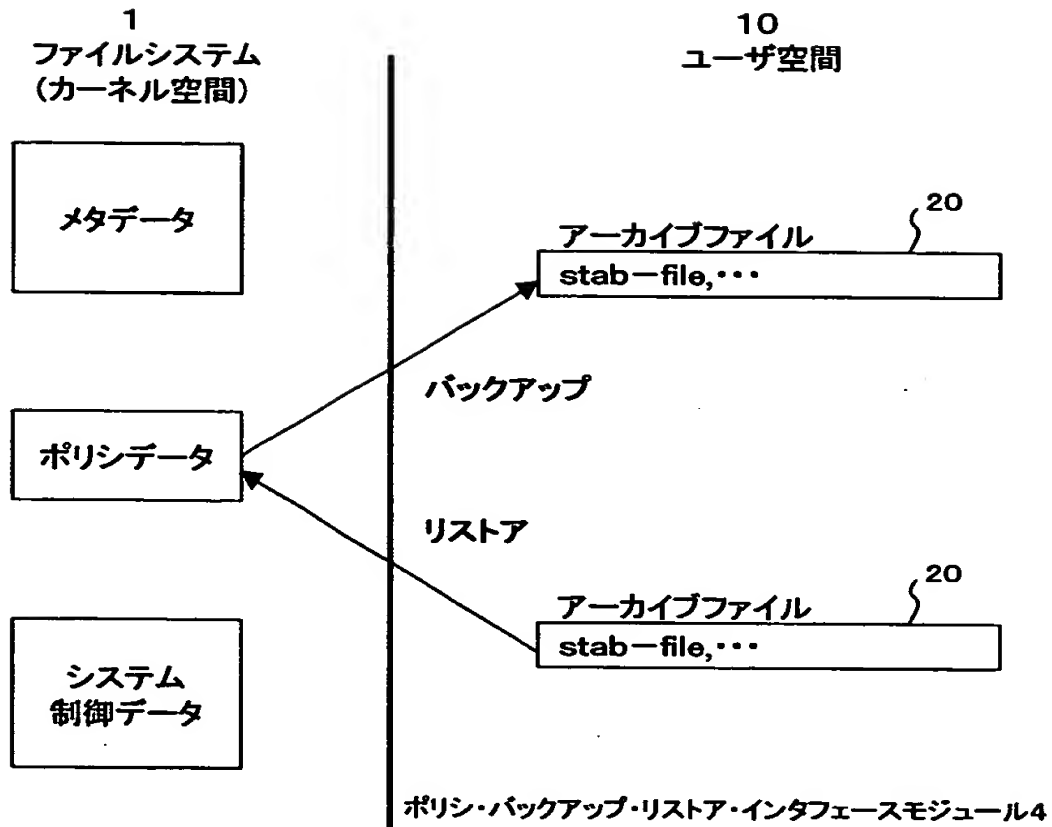
【図 2】



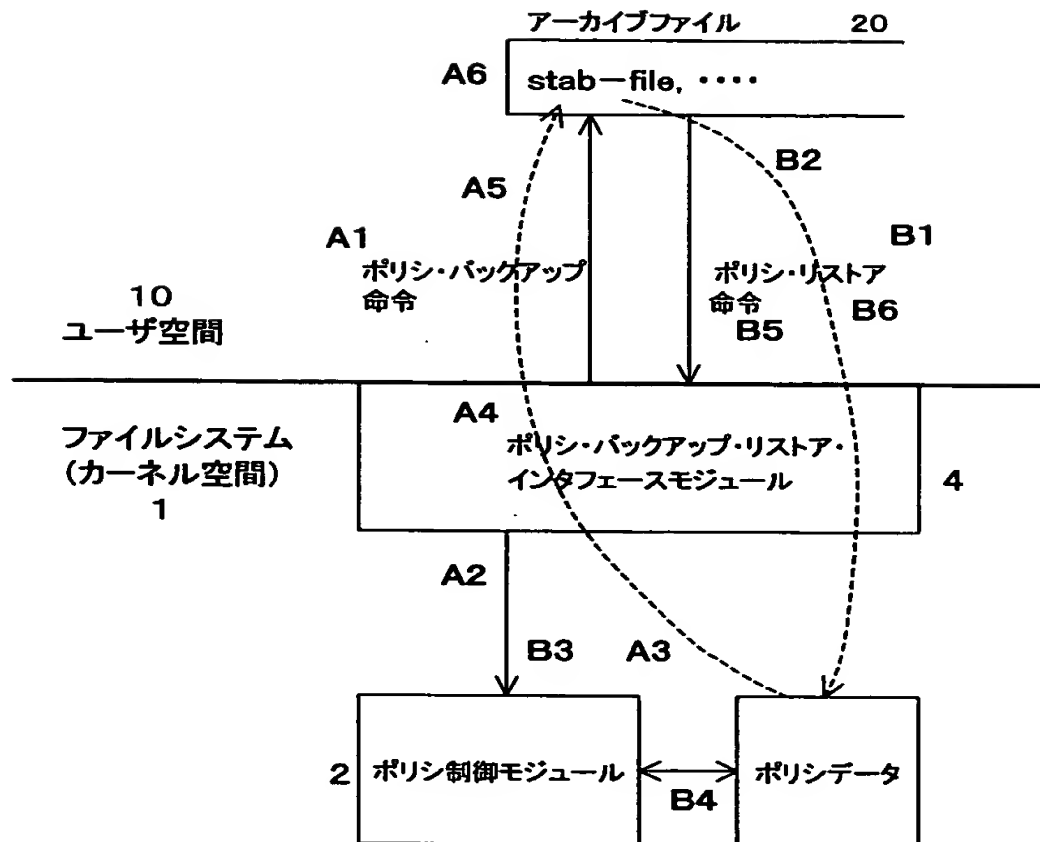
【図3】



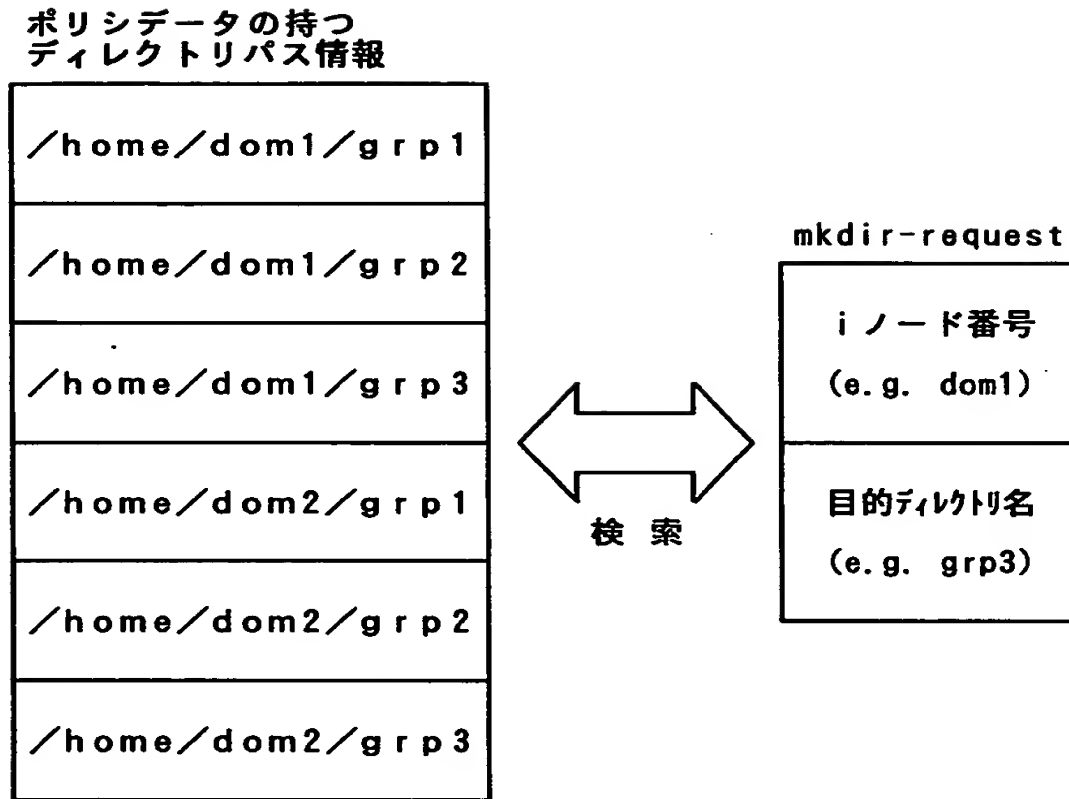
【図4】



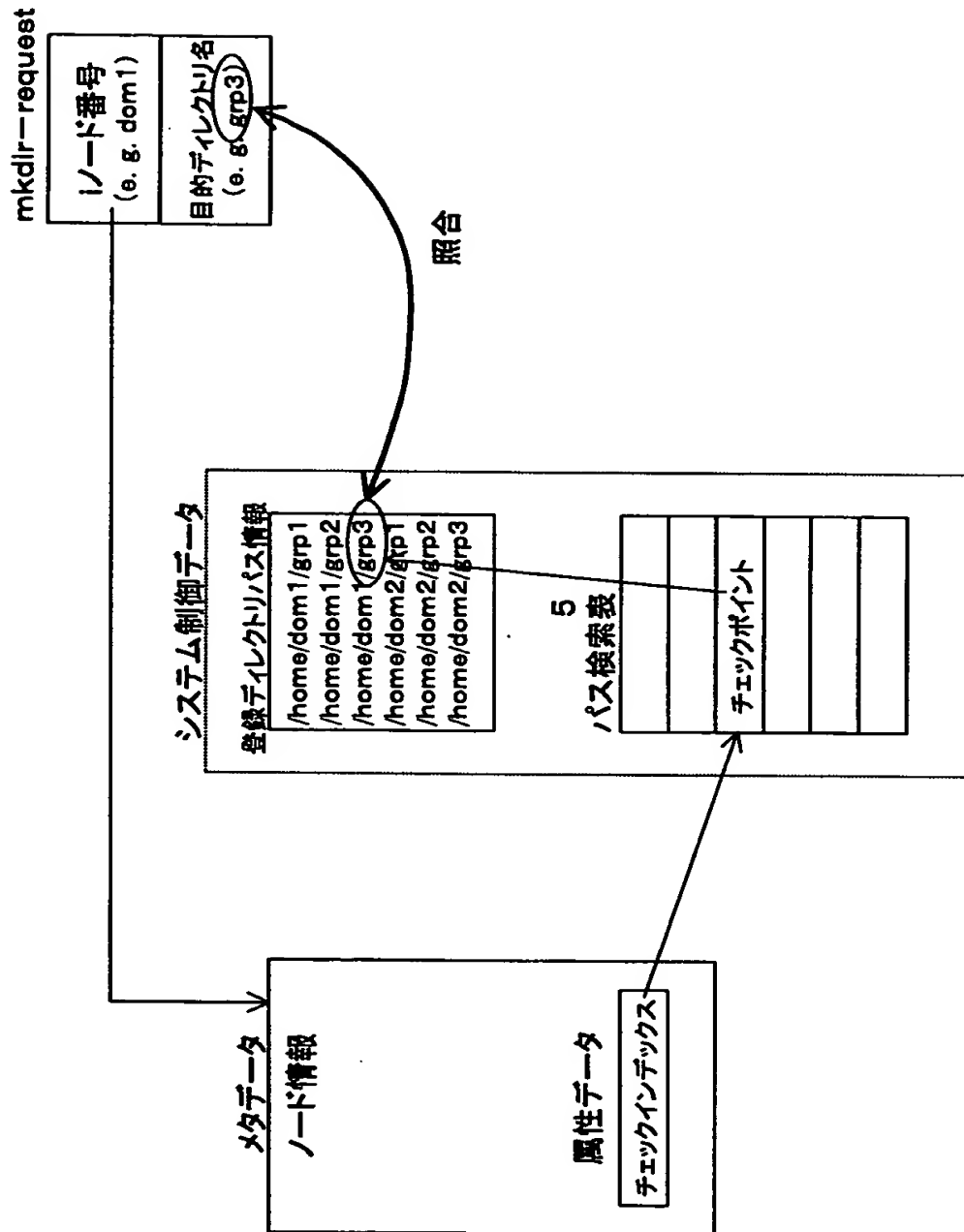
【図5】



【図 6】

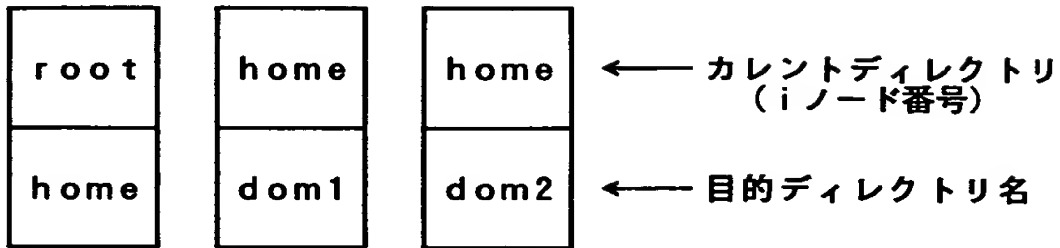


【図 7】

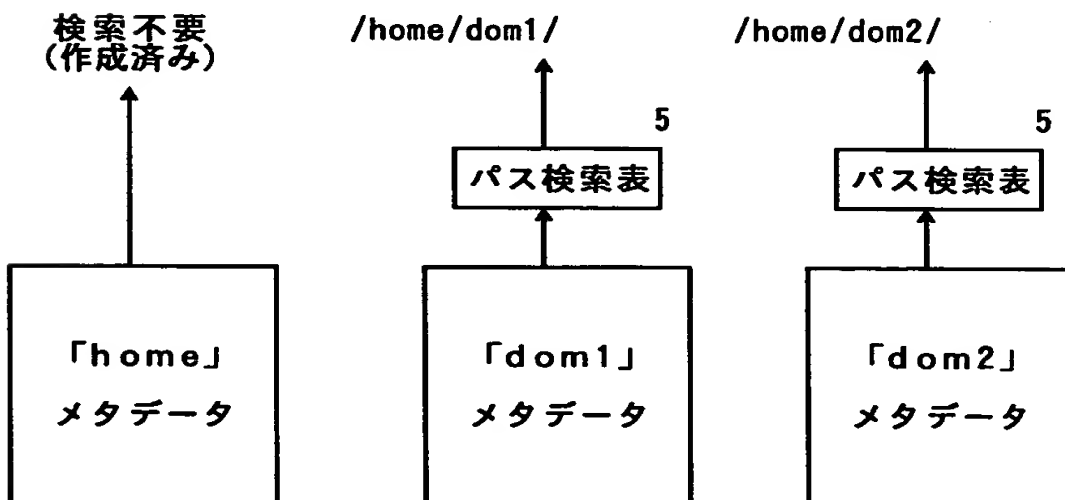
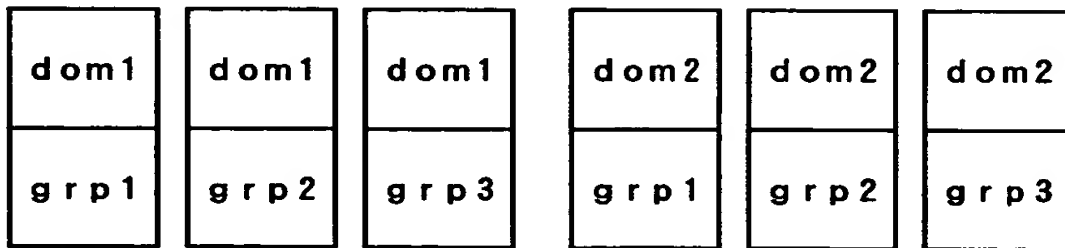


【図 8】

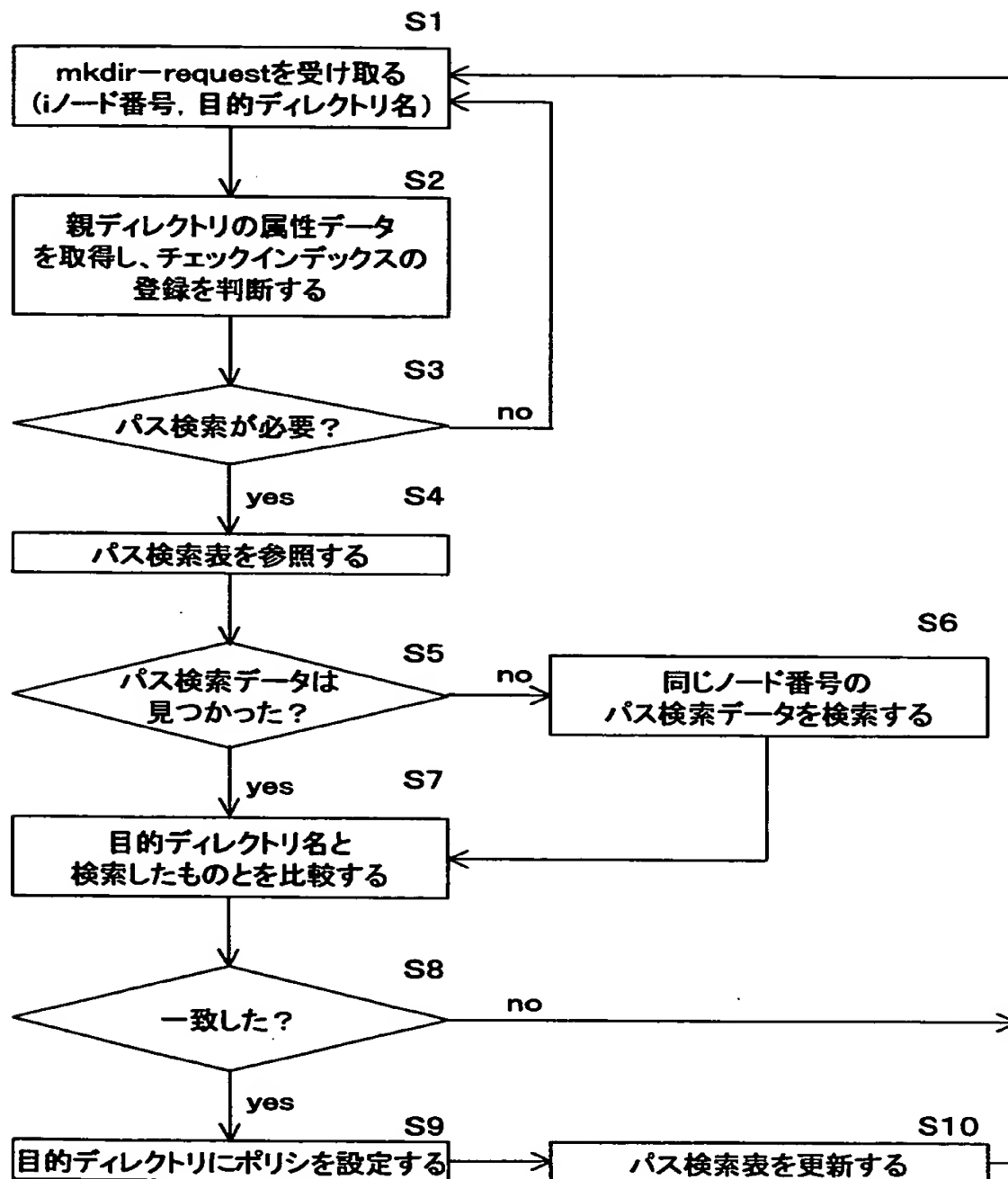
* 発行済み「メイク・ディレクトリ・リクエスト命令」



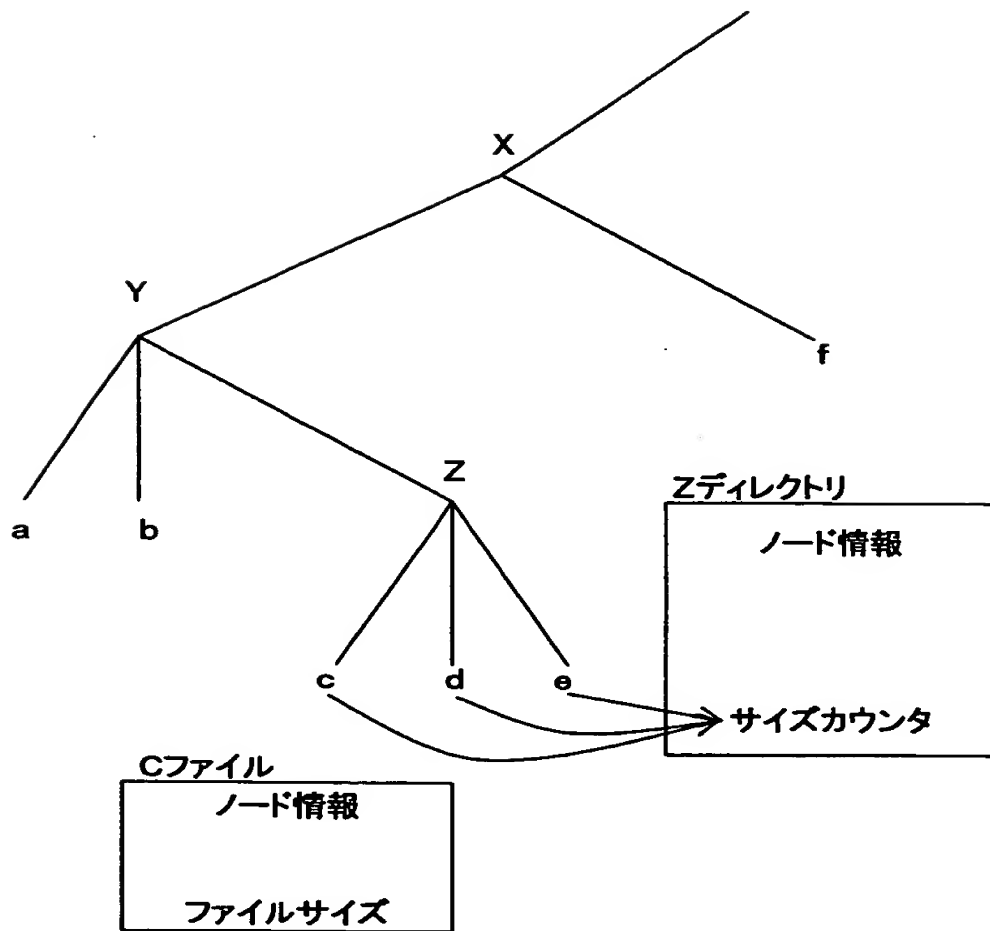
* 一部が未発行の「メイク・ディレクトリ・リクエスト命令」



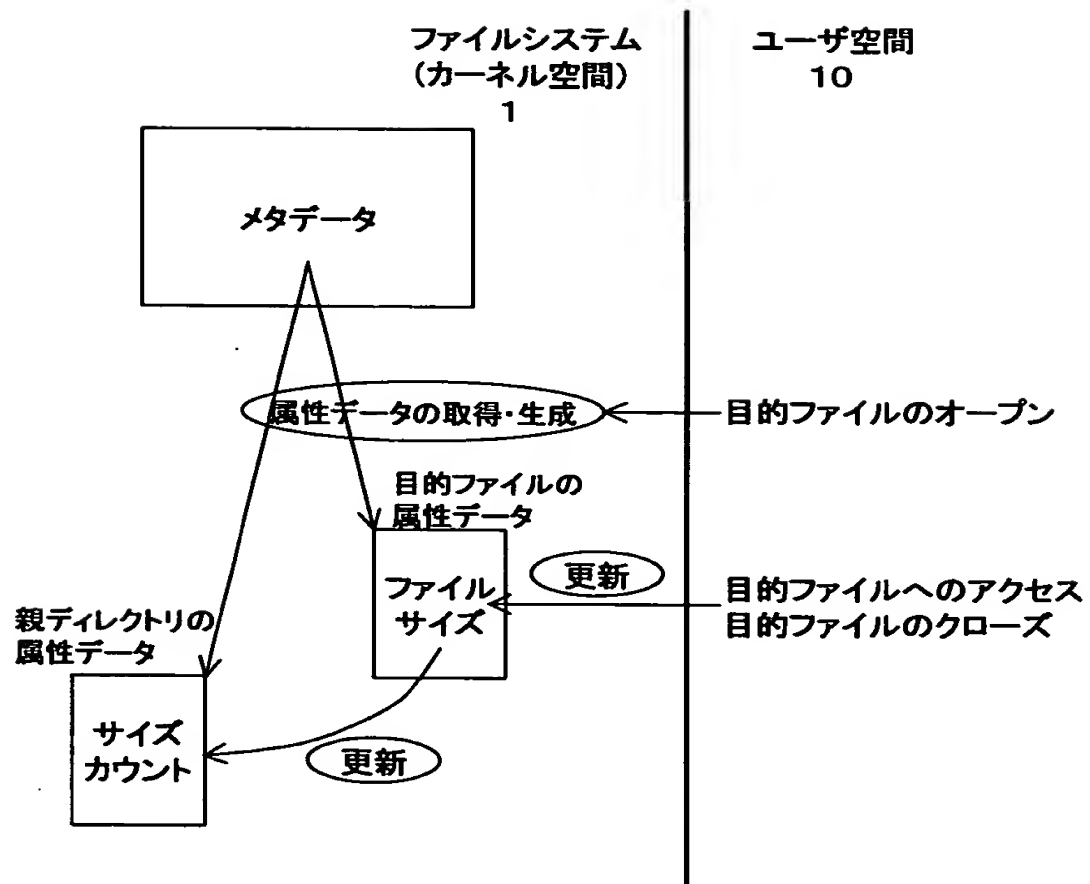
【図9】



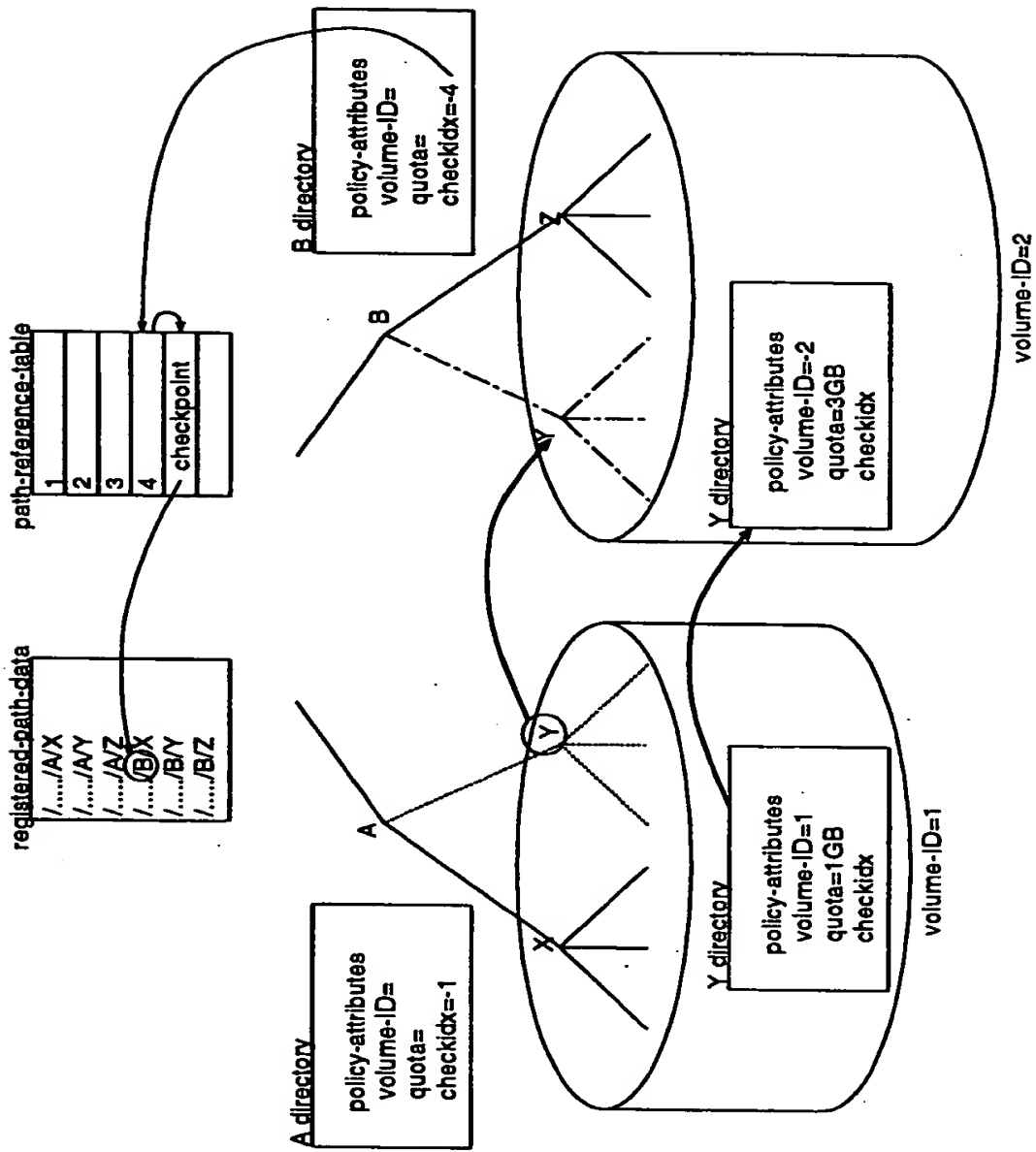
【図 1 0】



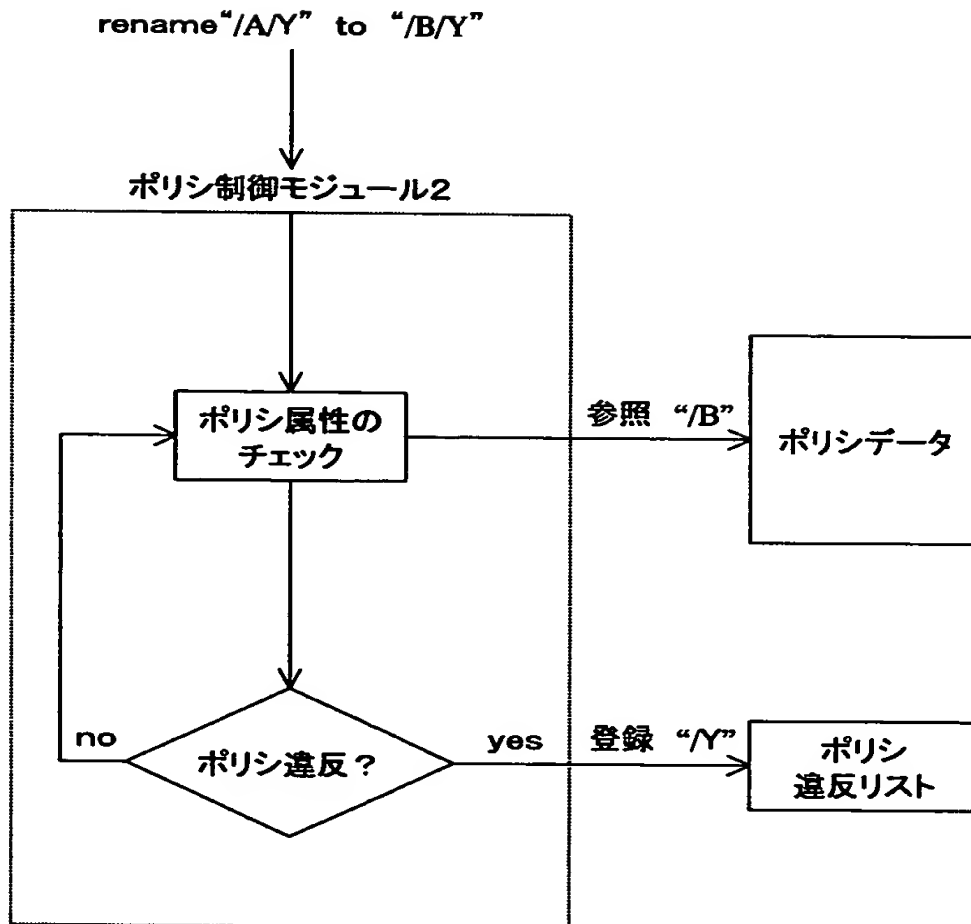
【図 11】



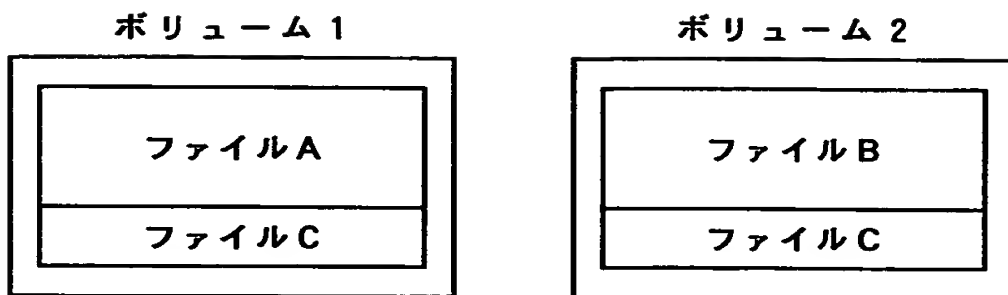
【図 12】



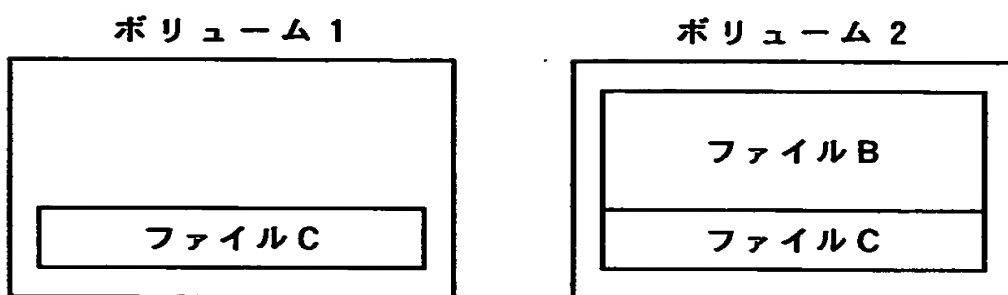
【図13】



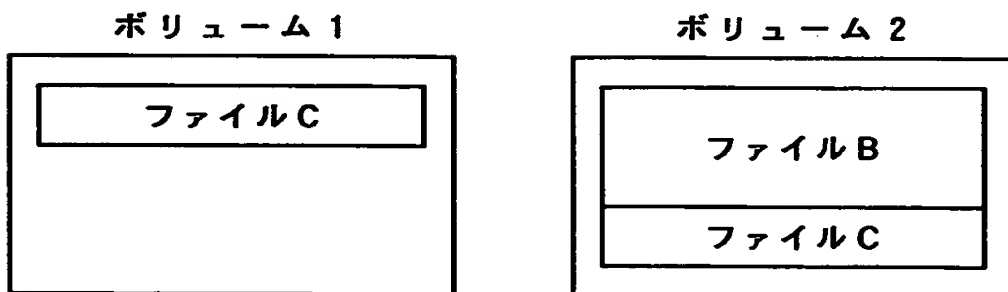
【図14】



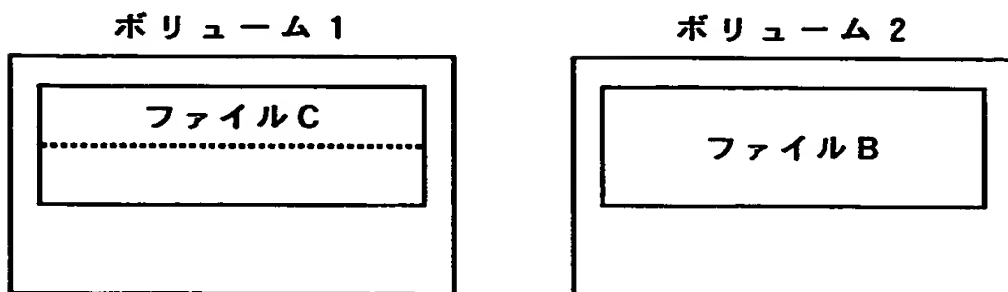
(a)



(b)



(c)



(d)

【書類名】 要約書

【要約】

【課題】 本発明は、通常のファイル操作インタフェースの性能を損なうことなく、また、通常のファイルシステムとの互換性を実現しつつ、固有の属性を付加することを實現する新たなファイルシステムの提供を目的とする。

【解決手段】 ディレクトリのパス情報とそれに対応付けて指定される 1 つ又は複数の属性データとで構成されるポリシデータを入力する手段と、入力されたポリシデータに従ってディレクトリに割り付けた属性データについては、その配下のディレクトリに継承させる形で割り付けるとともに、継承を要求されない属性データについては、入力されたポリシデータの持つ属性データを優先する形でディレクトリに割り付けることで、各ディレクトリに属性データを割り付ける手段とを備えるように構成する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日
[変更理由] 住所変更
住 所 神奈川県川崎市中原区上小田中4丁目1番1号
氏 名 富士通株式会社